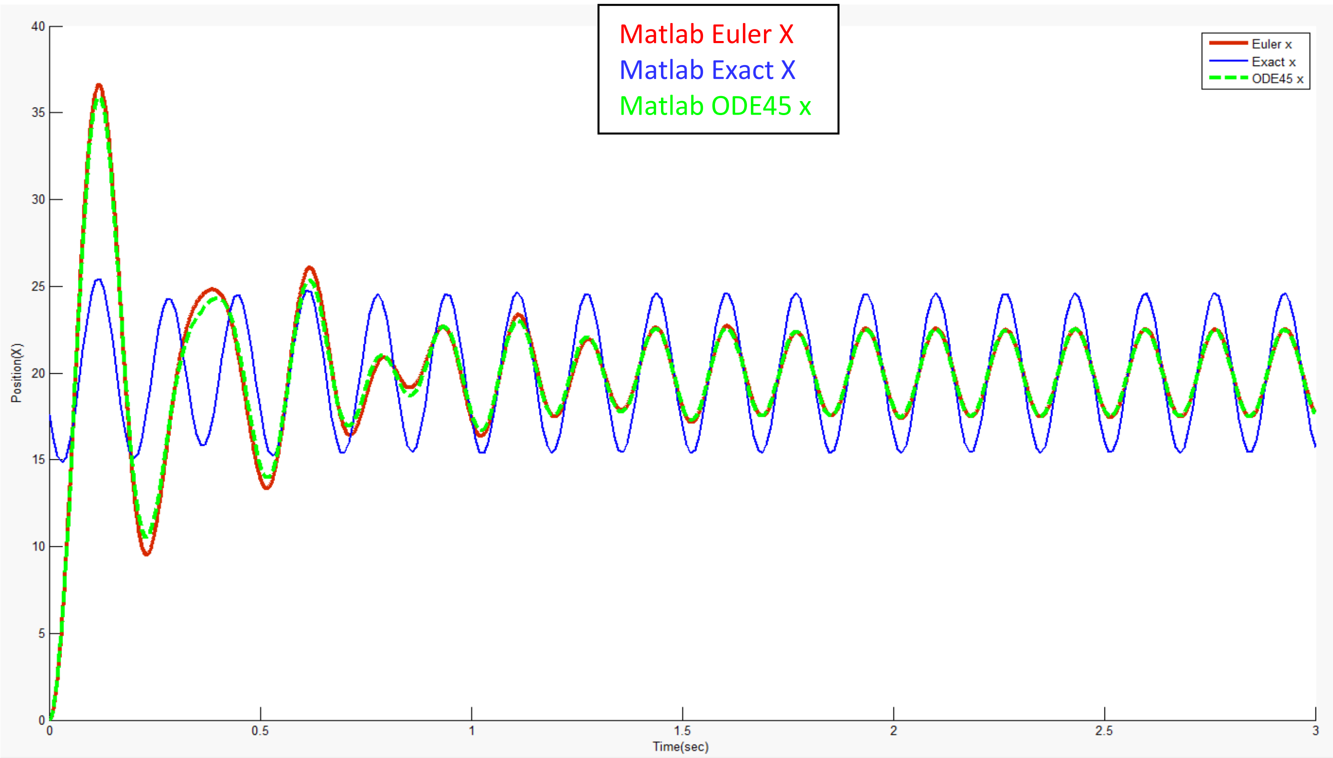
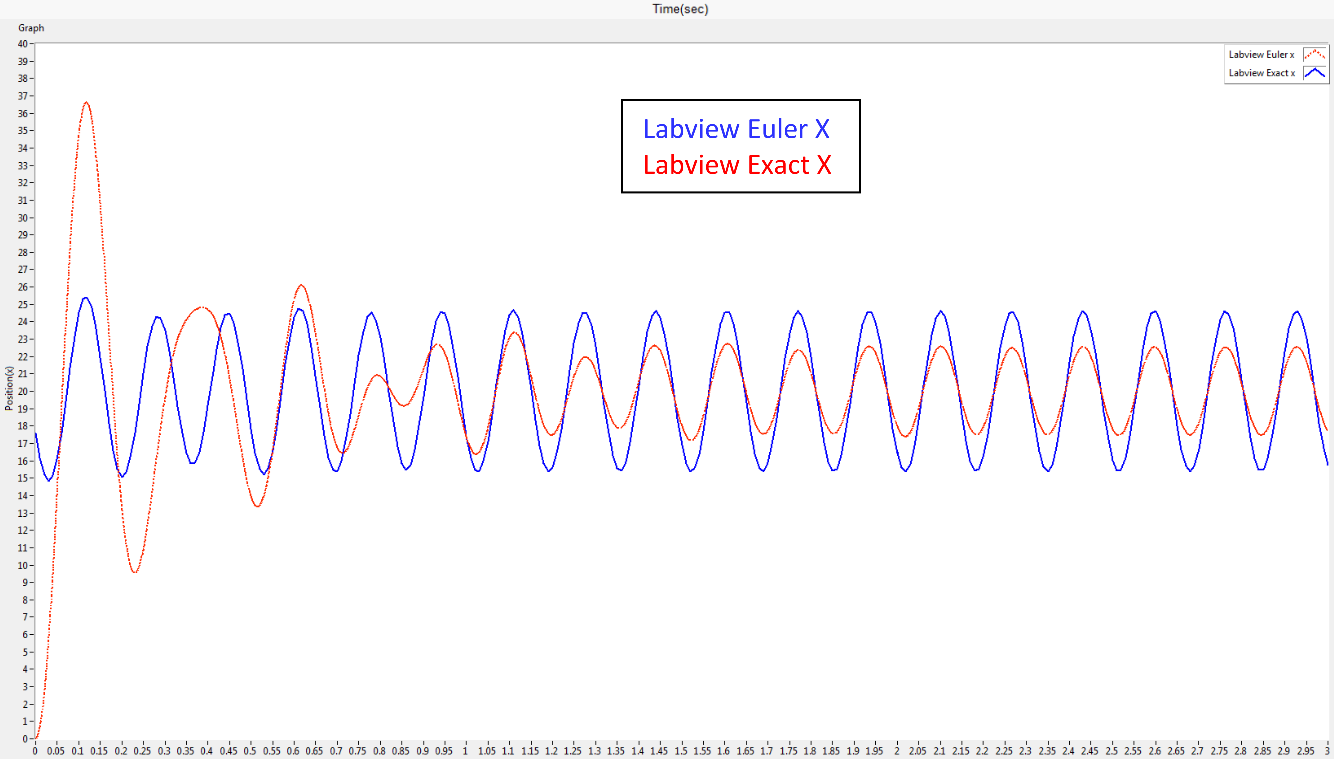
Clair Cunningham Homework #3 Numerical Methods

Second Order Linear Ordinary Differential Equation

The solutions obtained all had consistent after periods after 1.5 seconds within each plot and between plots. The peak and trough values varied though in where the Exact answer had greater value than the rest and Euler/ODE45 were the same smaller values. Also, the Exact plot reached a steady state much faster than the other two plots; therefore, the exact solution should be desired when dealing with small times smaller than 1.5 seconds and or when the true peak is desired.





```
%% Euler
%Clair Cunningham      Homework #3      Numerical Methods
%Clear previous runs
close all; clear all; clc;
%base equation: a = -2*damp*nfreq*v - nfreq^2*x + nfreq^2*f(t);
%set up constants
%final_time = 3 sec;
t = 0:.001:3;
h = 0.001; %time step
A = 20;
B = 3.4;
freq = 38;
damp = 0.12;
nfreq = 25;
%set up initial conditions:
x(1) = 0;
v(1) = 0; %xdot = v;
%f(1) = A+B*sin(freq*0); %f(t) = A+B*sin(freq*t);
%a(1) = nfreq^2*f(1)-2*damp*nfreq*v(1) - nfreq^2*x(1);

%initiate for loop since end is known
for i = 1:3000
    f(i) = A+B*sin(freq*t(i));
    a(i) = nfreq^2*f(i) - 2*damp*nfreq*v(i) - nfreq^2*x(i);
    v(i+1) = v(i) + h*a(i);
    x(i+1) = x(i) + h*v(i);
end

figure(1)
hold on;
plot(t,x);
legend ('Euler x');
ylabel('Position(X)'); xlabel('Time(sec)');
```

```
%% Exact
%Clair Cunningham   Homework #3 Numerical Methods
%Uses the exact equation solution given
%Clear Command Window and variables
clear all; clc;
%create known constants
A = 20;
B = 3.4;
damp = 0.12;
freq = 38;
nfreq = 25;
%solve for constants assuming variables are constant;
delta = ((2*damp*freq/nfreq)^2+(1-(freq/nfreq)^2)^2)^(1/2);
freqnd = nfreq*sqrt(1-damp^2);
phi = atan2((2*damp*freq/nfreq),(1-(freq/nfreq)^2));
lambda = atan2((-freqnd*(A+delta*B*sin(-phi))),(-damp*nfreq*A-damp*nfreq*delta*B*sin(-phi)-delta*B*freq*cos(-phi)));
c = asin(-A-delta*B*sin(-phi))/lambda;
t = 0:.01:3;
%solve for positions
for i=1:301
    x(i) = c*exp(-damp*nfreq*t(i))*sin(freqnd*t(i)+lambda)+A+delta*B*sin(freq*t(i)-phi);
end
figure(1)
plot(t,x)
legend('Euler x','Exact x');
```

```
function [t,data] = Homework_3_ODEGRAD45()
%Clair Cunningham      Homework #3      Numerical Methods
%Uses built-in Matlab ODE45 function to solve
%Clear previous runs
clear all; clc;
%base equation: a = -2*damp*nfreq*v - nfreq^2*x + nfreq^2*f(t);
%where f(t) = A+B*sin(freq*t)
%set up initial conditions
    x0 = 0;
    v0 = 0;
%Create Time and Data vectors for use in ode45
    wICvector = [x0,v0];
    tspan = [0 3];
%solve the ode problem fetching data from another equation
    [t,data] = ode45(@wdoteqns, tspan, wICvector);
%Define data vectors
    x = data(:,1);
%plot data
    figure(1)
    plot(t,x);
    legend ('Euler x', 'Exact x', 'ODE45 x');
end

function[wdotvector] = wdoteqns(t,w)
    %set up constants
    %seconds interval
    A = 20;
    B = 3.4;
    freq = 38;
    damp = 0.12;
    nfreq = 25;
    %define equations
    f = A+B*sin(freq*t);
    wdot1 = w(2);
    wdot2 = nfreq^2*f - 2*damp*nfreq*w(2) - nfreq^2*w(1) ; % acceleration
    wdotvector = [wdot1;wdot2];
end
```